**Laboratorio #4**

*Version #2*

Ejercicio #1

*Programa adjunto: e1.c*

*Ejecutando el programa:*

A screen shot of a computer program

Description automatically generated

Ambos procesos se encuentran en el estado en el que están "intentando adquirir" recursos, lo que sugiere que ambos están bloqueados esperando que el otro libere el recurso que necesitan. En este caso, el proceso 1 ha adquirido el recurso 1 y está esperando el recurso 2, que está bloqueado por el proceso 2, y viceversa. Por lo tanto, ambos procesos están en un estado de deadlock, lo que significa que ninguno de ellos puede continuar y el programa queda atrapado en un bucle infinito de espera.

*Describe cómo funciona un algoritmo de detección de deadlock y cómo se relaciona con la  
concurrencia en sistemas operativos.*

* Los algoritmos de detección de deadlocks son mecanismos utilizados por los sistemas operativos, que examinan el estado actual del sistema, específicamente los recursos asignados y las solicitudes pendientes que hay, y con esta información determinan si existe un ciclo en el grafo de asignación de recursos. Cuando se encuentra un ciclo, se determina que hay un deadlock.
* Esto se relaciona con la concurrencia en los sistemas operativos ya que es la capacidad de ejecutar múltiples procesos simultáneamente, sin caer en problemas como los deadlocks, que son una preocupación imporante en los sistemas concurrentes ya que estos pueden ocurrir cuando varios procesos están compitiendo por el mismo recurso, y al darse, se termina en una pérdida de productividad o incluso la muerte del sistema completo, por el bloqueo.

¿Qué estrategias podrían implementarse para prevenir deadlocks en sistemas concurrentes más  
complejos que el ejemplo proporcionado?

Explica cómo se podría modificar el código para introducir una situación de interbloqueo más sutil  
que no sea tan evidente como la inversión de recursos.

¿Qué métodos de detección y resolución de deadlocks conoces y cómo se aplican en sistemas  
operativos modernos?